We launch a Format Attack on the memory of write.c by passing it a malicious format string. In order to change the balance variable to 1234470, I ran the command

write $(printf “\x2c\xa0\x04\x08”).%#08x.%#08x.%#12x.%hnn. The address of the balance variable started at 0x0804a02c and was a four-byte int, therefore ending at 0x0804a02f. The format parameter %hhn enables us to overwrite the (one) least significant byte of the balance variable, which is pointed to by \x2c\xa0\x04\x08 in the printf command. The parameter writes the number of formatted bytes, which is (4 \* ‘.’ = 4) + (2 \* 08x = 16) + (12x = 12) + (3 \* “0x” = 6) = 38. The balance is initialized to 0x0012d687 and gets changed to 0x0012d626, since 26 is the hex value of 38. When run on the jaguar server, the output of this attack is

Safe: ,.%#08x.%#08x.%#08x#12x.%hhn

Vulnerable: ,.0xffd0deb0.0x000400. 0x16ad88

\*\*\* balance @ 0x8040a02c = 1234470 0x0012d626

This attack illustrates the need to format input before printing or using it, instead of allowing it to be used as a format string.